

Article

Data-Driven Geofencing Design for Point-of-Interest Notifiers Utilizing Genetic Algorithm

Iori Sasaki , Masatoshi Arikawa , Min Lu , Tomihiro Utsumi  and Ryo Sato

Graduate School of Engineering Science, Akita University, Akita 010-8502, Japan; arikawa@ie.akita-u.ac.jp (M.A.); lu@ie.akita-u.ac.jp (M.L.); tutsumi@ie.akita-u.ac.jp (T.U.); sato_ryo@ie.akita-u.ac.jp (R.S.)

* Correspondence: sasaki@lab.akita-u.info

Abstract: This study proposes a method for generating geofences driven by GPS trajectory data to realize scalable point-of-interest (POI) notifiers, encouraging walking tourists to discover new local spots. The case study revealed that manual geofence settings degrade the location relevance and user coverage—key objectives of POI notifiers—and hinder the scalability and reliability of services. The formalization presented computationally equips geofence designers with practical solutions through two implementations based on prior GPS trajectory logs: (1) a multiobjective genetic algorithm that suggests cost-effective geofences by providing trade-off visualizations and (2) a user coverage-penalized genetic algorithm that determines an optimal geofence based on the designers' expectations. The feasibility and stability of the proposed implementations were tested in areas with varying tourist flow patterns. A comparative survey among manual settings, settings incorporating a reliability simulation, and data-driven settings demonstrates significant performance improvements for geofence services.

Keywords: geo-notification; geofencing; GPS trajectory; flexible circular geofence; genetic algorithm; trigger event separation



Citation: Sasaki, I.; Arikawa, M.; Lu, M.; Utsumi, T.; Sato, R. Data-Driven Geofencing Design for Point-of-Interest Notifiers Utilizing Genetic Algorithm. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 174. <https://doi.org/10.3390/ijgi13060174>

Academic Editors: Wei Huang and Wolfgang Kainz

Received: 27 February 2024

Revised: 21 May 2024

Accepted: 23 May 2024

Published: 25 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

This study addresses the topic of geo-notifications (also referred to as location-based messages or location-dependent messages in [1]), which enhance tourists' mobile experiences in a city. With the development of GPS-enabled devices, a variety of location-based services (LBSs) such as pedestrian navigation, local searches, and driving assistance have emerged [2]. These services not only meet user demands based on pull services—where users actively request information—but also utilize push services to proactively deliver relevant information to users based on predefined criteria or events by sensing and recognizing users' context in real time. To advance such proactive SBSs, many businesses apply geofencing technology to their service designs. Geofences are virtual boundaries that represent the vicinity of points of interest (POIs) or define arbitrary spatial areas. Geofencing monitors user locations, detects their entry and exit from geofences, and provides corresponding services. This concept is utilized in various fields, for example, not only for human mobility, such as healthcare monitoring [3], dockless shared bikes [4], and task reminders [5], but also for unmanned aircraft systems [6] and animal protection [7]. Regional marketing also recognizes the value of location-aware tourist experiences [8,9]. Geo-notifications can leverage the effectiveness of geofencing technology to send messages about product offers [10], coupons [11], and customer surveys [1] at appropriate times.

Due to the application programming interfaces developed for Android by Google LLC (Mountain View, CA, USA) [12] and for iOS by Apple Inc. (Cupertino, CA, USA) [13], application developers and service designers can easily implement geofence applications. Nevertheless, constructing geofence applications encounters many difficulties in computation, management, and design. The trade-off between geofence reliability and energy

efficiency can be critical for stable computations of geofence services. Determining whether a point lies within a geofence has been a significant focus in research, particularly for introducing numerous polygonal low-cost geofences. Novel algorithms based on the quad-tree data structure [14], hybrid hashing scheme [15], trajectory partitioning [16], and the triangle weight characterization with adjacency in three-dimensional environments [17] have been utilized in these studies. In addition, infrastructure and frameworks are crucial for effective management. To allow service providers to manage their geofence applications continuously, prior research has proposed establishing a marketplace for geofence services with various purposes [18] and editors for the interactive creation of geofences [19]. Existing research on geofence designs, specifically, the configuration methods for their parameters, aims to ensure appropriate detection of the tourist context. Garg et al. estimated relationships between user location and affinity for products using a dataset that included e-commerce application logs and their locations, which they used for geofence settings [10]. Garzon et al. investigated the general problems of geofence settings from technical perspectives and proposed a method for calculating a reliability score [20]. Nevertheless, the exploration of this research area has been limited in terms of the diversity of user experiences. In addition, methods for computational determination of optimal geofences have not yet been established, which is a critical discussion for scalable geofence services. This study focuses on a type of geo-notifications in walking tours, which is as a typical scenario of geofence services in [18], and it aims to overcome the design problems.

The main contributions of this study are summarized as follows: (1) it formalizes the optimal conditions for POI notifiers, a type of geo-notification in tourism, based on their design problems; (2) it demonstrates that practical geofence parameters can be generated by using GPS trajectory data; and (3) it shows that data-driven geofences can be effective in terms of both stability and usability compared to conventional manual and simulation-based settings. Section 2 provides a detailed background on POI notifiers and outlines the problem definitions addressed in this research. Section 3 describes the computational methodology, i.e., the automatic geofence determinations, to resolve these problems, which includes a conceptual overview of geofence determination, formalization of objective functions, and algorithm implementations. The implementation design introduces two different approaches, utilizing multiobjective and penalized genetic algorithms. Using a synthetic GPS trajectory dataset, Section 4 examines the abilities of converging solutions in the proposed implementation in areas with varying tourist flow patterns. Furthermore, Section 5 presents a comparative survey among fully manual settings, settings with a reliability simulation, and data-driven settings to verify the geofence performance. This study extends our previous work at the 31st International Conference on Advances in Geographic Information Systems (ACM SIGSPATIAL 2023) [21], providing new discussion on an approach using a multiobjective genetic algorithm (refer to Section 3.3.1 for details) and experimental analysis in Sections 4 and 5.

2. Problem Definition of Geofencing Design

This section defines geofencing design by first describing the actual development of POI notifiers in Section 2.1. Through our case study and subsequent application developments, we explore the requirements for geofence determinations from both technical and service quality perspectives.

2.1. Background: POI Notifiers as a Local Search Service for Tourists

The POI notifier is a widely used example of geofence services in tourism [18]. Self-guided walking tours provide participants the freedom to explore the city, discover new spots, and experience the local nature and culture. A POI notifier supports this activity for tourists. While users are near POIs or strolling by them, it notifies them of their presence, ensuring that they do not miss out on the spots that tour designers would recommend them to visit.

To elucidate the role and use case of POI notifiers, this study introduces a mobile application developed by the authors, named *DJWalkers*, illustrated in Figure 1. Users need only press the play button before initiating their walking tours. During these walks, *DJWalkers* continuously tracks the location and monitors entry into two types of triggers for playing audio: *nearby-geofence* (red circle) and *staying-geofence* (blue circle). Initially, a user might enter a *nearby-geofence* associated with a specific POI, such as an Asian restaurant and historical monument, without realizing it. *DJWalkers* then automatically plays an audio prompt, such as “A new POI is near you. Do not miss it!” In addition, it provides brief information about the reason for visiting and the activities available there. If motivated by the audio, and the user decides to visit the POI, a *staying-geofence* that covers the area of the POI might subsequently detect their presence. During the user’s stay at the location or while viewing the monument, *DJWalkers* offers audio content detailing the site’s historical significance and untold stories. In essence, *nearby-geofences* serve as “encouragers” or “promoters,” while *staying-geofences* act as “educators” or “docents”. This study addresses the former type of geofence and explores its design challenges.

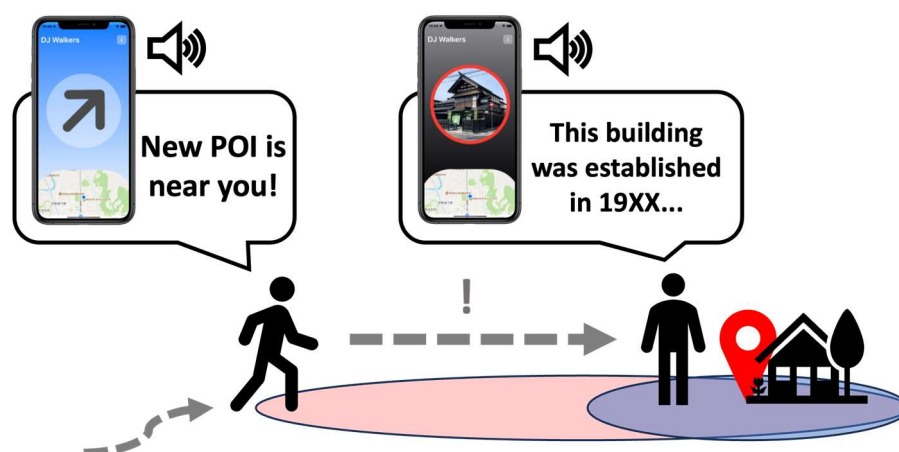


Figure 1. Two types of geofences implemented by *DJWalkers*, a geofence-based mobile application developed by the authors: (1) *nearby-geofences* (red circle), which encourage users to visit undiscovered POIs, and (2) *staying-geofences* (blue circle), which provide users who arrive at POIs with historical background information. In this study, POI notifiers are identified as the proactive service triggered by *nearby-geofences*.

2.2. Design Problems of POI Notifiers

This subsection details the design problems from various perspectives. Creating effective geofence services requires proficient IT experts who understand the technical constraints related to geofence settings and domain experts familiar with tourists’ movements within target regions. However, this may not always be feasible for businesses. The technical perspectives in Section 2.2.1 provide general insights into geofence design, supporting the necessity of a data-driven automation approach. The service quality perspectives in Section 2.2.2 are closely related to usability and define the unique guidelines for POI notifiers. This subsection leads to the formalization of objectives in Section 3.2.

2.2.1. Technical Perspectives

Domain experts often find it challenging to account for the technical limitations of geofence services [22]. For example, GPS accuracy is a critical factor in LBSs. In general, GPS-enabled devices record location data that include random noise due to surrounding obstacles and atmospheric effects. Being indoors can significantly reduce accuracy, potentially causing the degradation of LBSs. In addition, the sampling rate, which determines how frequently it is checked whether a user is inside or outside geofences, is also a delicate parameter that can affect the reliability of geofences [23]. Overlooking these features can

lead to inappropriate activations and deactivations, regardless of their appropriateness from service quality perspectives.

To enrich the user experiences of walking tourism over time, POI notifiers are expected to implement more POIs and corresponding geofences. This necessitates a discussion on the scalability issues of geofence services. Especially when many geofences are set in a small area, they may overlap and trigger multiple events simultaneously, leading to undesired operations of the application. Furthermore, setting a large number of practical geofences while considering both technical and service quality can be labor-intensive. This could prolong the geofence setup process, posing a significant challenge to scalability.

2.2.2. Service Quality Perspectives

Figure 2 presents a case study on geofence designs for POI notifiers. To ensure timely notifications, geofence designers analyze pedestrian areas to determine optimal locations for geofence placement.

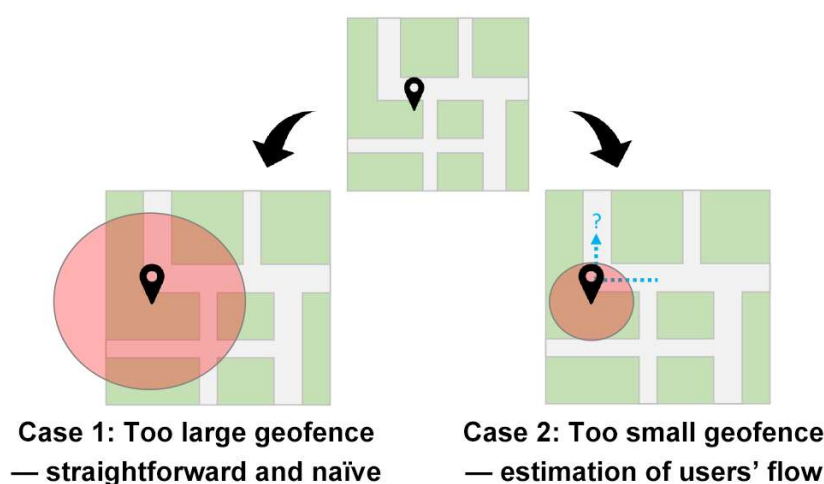


Figure 2. Case study on geofence settings for POI notifiers. In Case 1, using a larger geofence is straightforward and effective for covering more pedestrians but can lead to poor location relevance of notifications about the POI. In contrast, Case 2 estimates tourist flows and precisely covers the footpath to achieve high location relevance. However, if the flow predictions are inaccurate, this approach may significantly reduce user coverage, thereby missing critical interactions.

One person set a larger geofence, covering a more extensive area (Case 1 in Figure 2). This method effectively captures more people. In general, setting an overly large geofence can result in providing information about a place far from the user's current position. This may lead to tourists' inability to identify the location of the POI or reluctance to visit. Such a situation can cause frustration and confusion. Therefore, the distance between the POI and the position at which users receive notifications is related to the relevance of the information. This study refers to this inappropriate situation as a problem of poor location relevance in POI notifiers.

Another person estimates tourist flows using their heuristics and places a geofence of an appropriate size to capture those who pass through specific paths (Case 2 in Figure 2). This approach maintains relatively short for all users. However, it cannot guarantee high user coverage. This method may be easily influenced by designers' misunderstandings and the dynamics in the target areas. POI notifiers should employ geofences to capture as many pedestrians as possible to promote POIs effectively. Considering environmental factors, geofencing should aim to maximize the number of users covered.

Based on the previous discussion, a trade-off appears to exist between location relevance and user coverage. This study treats this trade-off as an optimization problem, aiming to determine a geofence that is not dominated by other potential solutions.

3. Methodology: Automatic Geofence Determination Utilizing Genetic Algorithm

To address the problems discussed in Section 2.2, this section introduces a methodology for automatic geofence determination. This study proposes using GPS trajectory data to determine circular geofence parameters, specifically, a center position and radius, within a defined space. Two objective functions are introduced: (i) location relevance and (ii) user coverage. The determination process can effectively be implemented using genetic algorithms.

3.1. Concept Set for Geofencing Optimization

To define the directions of our determination approach, Section 3.1 introduces three fundamental concepts:

- (1) Trigger event separation—for practical services, geofence designs must ensure the delivery of a single notification within parameter constraints or handle spatially unrelated processes for event selections. Section 3.1.1 provides a framework for consistent geofencing operations and declares the scope of our geofencing optimization.
- (2) Flexible circular geofence—given the potential for future scaling of POI notifiers and the feasibility of optimization, a simple and effective geofence representation is crucial. Section 3.1.2 involves defining geofence variables for optimization.
- (3) GPS trajectory data-driven determination—to consider technical and behavioral patterns, prior tracking data of user movements can serve as a feasible solution for geofence services to autonomously improve. Section 3.1.3 illustrates the data inputs in our proposed methods.

3.1.1. Trigger Event Separation

The strategy of trigger event separation clearly separates the computations of geofence services into a *spatial trigger module* and an *event selection module*. First, the *spatial trigger module* should allow for the detection of multiple geofences and a user's spatial context, for example, entry, exit, and dwell, simultaneously. After that, the *event selection module* involves determining a single event to deliver to users based on the results of the *spatial trigger module*. For example, the application selects preferable notifications for users by employing collaborative filtering techniques, which analyze the transactions and preferences of users to identify patterns and suggest content that appeals to similar users [24]. In other words, the *spatial trigger module* accepts overlaps of geofences, and the *event selection module* solves overlapping problems (refer to Section 2.2.1 for background). This study focuses only on the *spatial triggers* and attempts to optimize geofence parameters without spatial constraints arising from the positions of neighboring POIs. This concept allows POI notifiers to set many POIs even in small regions, which accelerates the scalability of POI notifiers.

3.1.2. Flexible Circular Geofence

Our determination process represents circular geofences (CGFs), which are defined by a center point and radius. While some applications introduce polygon-shaped geofences with ray-casting methods [25], CGFs offer more scalable and straightforward implementations because they can determine whether a user is inside or outside the area, and they can do so simply by calculating the distance between the center point and the user's location. Many POI-based geofence services adopt *fixed CGFs* that share the same center point as POIs. Thereby, domain experts can set geofences based on user proximity to POIs. However, the intuitive nature and simplicity of *fixed CGFs* cannot adapt to the environment around POIs and cause inappropriate activation and inactivation [26]. For example, the case in which rivers or trees surround a target POI forces designers to set unnecessarily large *fixed CGFs* to avoid covering areas that users cannot enter. In this paper, automatic optimization applies advanced geofences—that is, *flexible CGFs*, which are not constrained by the positions of POIs. It hypothesizes that applying *flexible CGFs* would achieve more desirable and optimal settings for POI notifiers.

3.1.3. GPS Trajectory Data-Driven Determination

This study aims to introduce a data-driven approach to optimize POI notifiers. As geofence services are basically utilized in GPS-enabled devices, collecting GPS trajectory data via networks can be a feasible way as a part of the lifecycle of POI notifiers. The optimization process incorporates the GPS datastore into the calculation of evaluation values. Because GPS tracking data reflect not only recent trends of user movements but also GPS errors depending on surrounding environments [27], this approach has the potential to capture the technical and behavioral patterns of GPS datasets and overcome the problems, as shown in Section 2.2.1. Therefore, our determination process requires a GPS dataset as the data input. A high enough number of trajectories with higher sampling rates (i.e., 1–16 s intervals) in target cities is needed for the dataset.

The experimental analyses in this study have introduced a synthetic GPS dataset for comprehensive technical surveys. Ideally, service designers are expected to collect the actual GPS tracking data from their application users. Alternately, some research has utilized open-source GPS datasets reflecting actual human mobility, such as GeoLife [28] and Singapore taxis [29]. In contrast, the use of synthetic GPS trajectories can establish controllable and large-scale benchmarks [14,30].

Our simulation settings involve scanning three areas, each within a circle centered on the corresponding POIs and with a radius of 75 m. The simulation utilized road network data from OpenStreetMap [31] to generate synthetic trajectories (refer to Appendix A for details). As the areas display distinct patterns of GPS tracking, setting the scanning range r_{scan} , one of the hyperparameters of the proposed genetic algorithms, to 75 m is effective in evaluating our methods while considering the dynamics of human mobility. Figure 3 illustrates the positions of the target POIs: (A) Arts Theater, (B) Kogetsu Lake, and (C) Yojiro-Inari Shrine, along with the density maps of each scanning space. In the area of POI (A), a single main road is clearly visible on the bottom-right side of the figure. The area of POI (B) depicts slightly complex clusters, primarily scattered from the bottom to top-left side. Finally, interpreting the density structure in the area of POI (C) is the most challenging, as it appears to present multiple clusters around the POI.

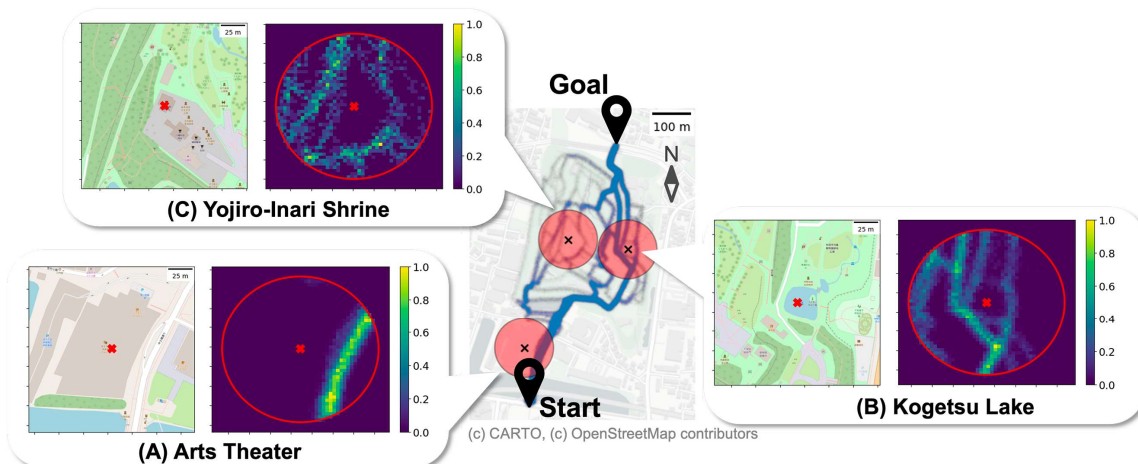


Figure 3. The simulation field is set in Senshu Park, located in the center of Akita City, Japan. The park features a walking trail surrounded by beautiful nature. Synthetic trajectory data points (blue), target POIs (x-mark), and scanning spaces with a 75 m radius (red circle) displayed in the figure are based on the EPSG:6691 coordinate system. This study selected three areas with different point patterns for GPS tracking: (A) Arts Theater, (B) Kogetsu Lake, and (C) Yojiro-Inari Shrine. The density values in heatmaps are normalized, ranging from 0.0 for the minimum value to 1.0 for the maximum value of each area. The area of POI (A) depicts the simplest patterns, while the area of POI (C) presents multiple and complex density structures.

3.2. Formalization of Objective Functions

As discussed in Section 2.2.2, location relevance and user coverage influence the service quality of POI notifiers. Section 3.2 converts these requirements into two evaluation functions, enabling computational determinations. Initially, following the defined optimization scope outlined in Section 3.1.1, these functions are dedicated only to spatial calculations. The parameters for geofences within these functions include a variable center position and radius, based on the *flexible CGF* model described in Section 3.1.2. Furthermore, the calculation of user coverage incorporates historical GPS log data, supporting the implementation of a data-driven optimization approach, as discussed in Section 3.1.3. The conceptual diagrams illustrating each objective are shown in Figure 4.

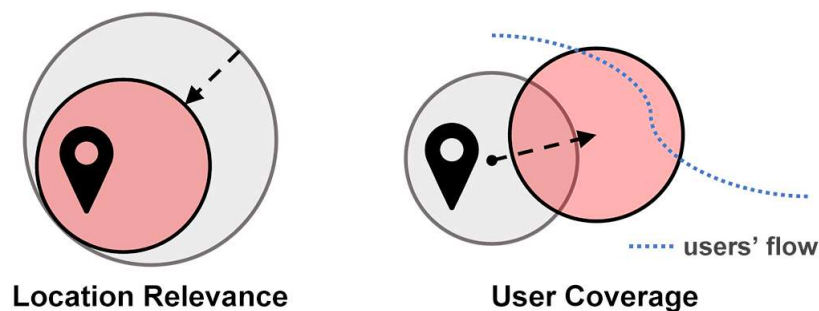


Figure 4. Conceptual diagrams of proposed objectives: **left**, location relevance, and **right**, user coverage. Gray circles represent nonoptimal geofences; the objectives can guide them to desirable geofences, that is, red circles.

3.2.1. Location Relevance

The distance to the destination and its information relevance can be proportional. Notifications about unnecessarily far places from current user locations should be avoided. This requires minimizing the distances from the boundaries of geofences to POIs.

Let $dist(poi, cgf.xy)$ be the Euclidean distance from the location of a POI defined as poi to the center position of the CGF denoted as $gf.xy$. Furthermore, $cgf.r$ stands for the radius of the CGF. Consequently, location relevance can be articulated as an average distance from the boundary of the geofence to a target POI. Given the computational cost and the need for scalability, the location relevance is defined simply by calculating the average of the largest distance $dist(poi, cgf.xy) + cgf.r$ and the shortest distance $|dist(poi, cgf.xy) - cgf.r|$. This paper attempts to minimize this value.

$$\frac{dist(poi, cgf.xy) + cgf.r + |dist(poi, cgf.xy) - cgf.r|}{2} \quad (1)$$

When considering the evaluation using this objective, the effects of arbitrary POI positioning can be crucial. Typically, service designers can be free to set the position of a POI as the reference point for intended spots. The consistency of output geofence parameters seems not to be guaranteed due to this arbitrariness. Nonetheless, our approach aims to minimize the distance between POIs and notification points, which inherently suggests that POIs should be positioned at points where users are most likely to arrive. This alignment ensures that desirable notifications can be accomplished.

3.2.2. User Coverage

To avoid setting geofences in places where few users cross or cannot enter due to walls and rivers, the number of users who receive notifications about the spot should be maximized.

Within a dataset of trajectories denoted by $D = \{Tr_1, Tr_2, \dots, Tr_m\}$, each trajectory Tr_j consists of a set of GPS point data represented as $Tr_j = \{p_{j_1}, p_{j_2}, \dots, p_{j_n}\}$. Let $p_{j_k} = (id_j, x_{j_k}, y_{j_k})$ depict a GPS point data; then, the user number can be defined as follows:

$$\left| \left\{ id_j \mid (x_{j_k} - cgf.x)^2 + (y_{j_k} - cgf.y)^2 \leq cgf.r^2 \right\} \right| \quad (2)$$

3.3. Implementation of Data-Driven Geofencing Design

To explore implementation methods of our data-driven geofencing design for service designers, this section presents two different optimization approaches:

- (1) Multiobjective genetic algorithm—Section 3.3.2 treats the two criteria as independent objective functions and attempts to optimize them equally.
- (2) User coverage-penalized genetic algorithm—Section 3.3.3 minimizes the location relevance value while treating user coverage as a soft constraint.

These approaches provide different methods for decision-making in geofence service developments. Section 3.3.1 illustrates the general workflow from data input by a service designer to geofence determination for the two proposals.

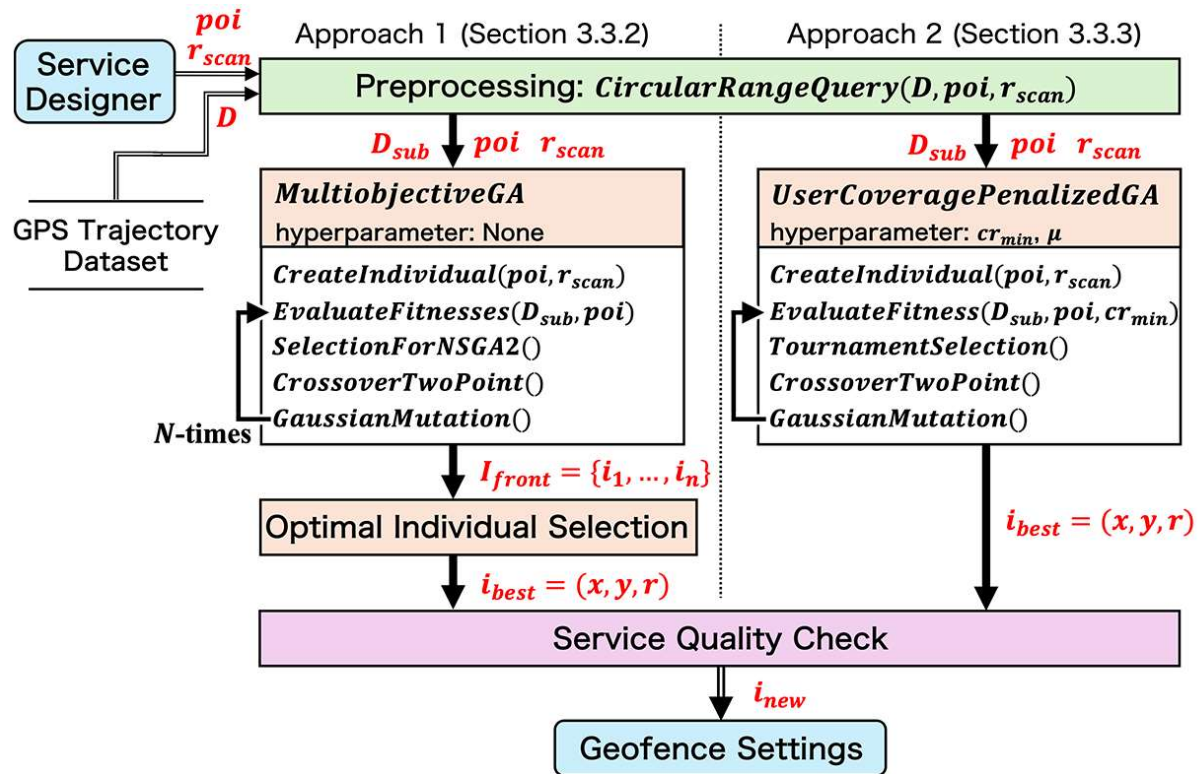
3.3.1. Workflow of Geofence Design

Figure 5 depicts the workflow for our data-driven geofence design, with data flow illustrated using red symbols for working data. This process assumes that a GPS trajectory dataset D was constructed by different systems, such as tourist tracking data accumulated on a server via mobile applications. Initially, service designers input the position of a POI, poi , corresponding to the target geofence to be optimized. The scanning range r_{scan} is also an input parameter defining the scanning space for our optimization. The general workflow can be divided into three phases: (i) preprocessing, (ii) geofence determination, and (iii) service quality check. In the preprocessing phase, the *CircularRangeQuery*(D, poi, r_{scan}) extracts a subtrajectory dataset, D_{sub} , which exists within the scanning range. The scanning range is defined as the circle region $(x - poi.x)^2 + (y - poi.y)^2 \leq r_{scan}^2$. In the geofence determination phases, genetic algorithms are applied to the subtrajectory dataset D_{sub} , which is detailed in Sections 3.3.2 and 3.3.3. As a result, the best geofence solution i_{best} is obtained. Service designers confirm the parameters based on the utility for their services and update the geofence settings i_{new} . This process is repeated for the number of geofences to be set. In this study, genetic algorithms were implemented using the hyperparameters listed in Table 1 and partially with DEAP, a Python library [32].

Table 1. Hyperparameters for our genetic algorithms used in all experiments described in this paper. These parameters have been determined to be optimal based on our preliminary experiments. Note that the minimum user coverage rate cr_{min} is not constant in this study and is not included in the table.

	Multiobjective Genetic Algorithm (Refer to Section 3.3.2)	User Coverage-Penalized Genetic Algorithm (Refer to Section 3.3.3)
μ	Not applicable	2000
Crossover probability	0.9	0.5
Mutation probability	0.3	0.1
Generation number	300	150
Population number	300	300

Note: in addition to specific parameters for the genetic algorithm, penalty coefficient μ was introduced for the user coverage-penalized genetic algorithm.



● Legend for Working Data

poi: The position of POI **r_scan**: The scanning range **D**: A GPS trajectory dataset
D_sub: A subset of **D** **I_front**: Candidates of geofence solution
i_best: A geofence solution **i_new**: A new geofence for **poi**

● Legend for Functions

CircularRangeQuery(D, poi, r_scan): Extract subtrajectories within the scanning range.
MultiobjectiveGA(D_sub, poi, r_scan): Execute multiobjective genetic algorithm.
UserCoveragePenalizedGA(D_sub, poi, r_scan): Execute single-objective genetic algorithm with penalty on the user coverage rate.

Figure 5. The workflow and data flow for the automatic geofence design process, from data input by a service designer to geofence determination, are illustrated in the diagram. The external entity is highlighted in blue. The workflow consists of three main phases: preprocessing (green), geofence determination (orange), and service quality check (pink). Working data are illustrated using red symbols. The processes of the genetic algorithms (Approaches 1 and 2) are detailed in Sections 3.3.2 and 3.3.3.

3.3.2. Multiobjective Genetic Algorithm

The multiobjective genetic algorithm approach simply treated two formalizations as independent objective functions and attempted to optimize them equally. Consequently, a Pareto front is visualized for geofence designers, illustrating the trade-offs between the objective functions. In the next step, designers need to select a preferred solution from multiple geofence candidates (Optimal Individual Selection described in Figure 5).

The implementation of the multiobjective genetic algorithm employs the Non-dominated Sorting Genetic Algorithm II (NSGA-II) that is well known for its effectiveness in handling multiobjective optimization problems [33]. In the context of this study, it helped in simultaneously optimizing both location relevance and user coverage objectives. The

MultiobjectiveGA in Figure 5 illustrates the flow of the proposed optimization method using NSGA-II. Following this, NSGA-II is executed to determine a Pareto front for the desired geofence parameters. The basic flow is similar to a single-objective genetic algorithm, as applied for the second approach in Section 3.3.3; however, it is unique in its selection methods—non-dominated sorting and crowding distance calculation. First, the process begins by generating a set of candidate CGFs, called a population in the first generation. As for an individual geofence $i = (x, y, r)$, the center point is converted from two uniform random numbers defining a polar coordinate, and the radius is also determined by a uniform random number set to be within the scanning range (refer to Algorithm 1). Second, the evaluation of every value of i is conducted by the two following objective functions $f_1(i)$ and $f_2(i)$ based on Section 3.2 for each individual. Each of them is normalized by the scanning range value r_{scan} and the number of users in a GPS subtrajectory dataset D_{sub} .

$$\text{Minimize } f_1(i) = \frac{\text{dist}(\text{poi}, i.xy) + i.r + |\text{dist}(\text{poi}, i.xy) - i.r|}{2r_{scan}} \quad (3)$$

$$\text{Maximize } f_2(i) = \frac{\left| \left\{ p_k.id \mid (p_k.x - i.x)^2 + (p_k.y - i.y)^2 \leq i.r^2, p_k \in D_{sub} \right\} \right|}{\left| \{ q_j.id \mid q_j \in D_{sub} \} \right|} \quad (4)$$

According to these indicators, NSGA-II executes non-dominated sorting. It categorizes the solutions into different domination levels. The first level comprises solutions that are not dominated by any other solution; the second front includes solutions only dominated by those in the first front. To ensure diversity among the solutions, the crowding distance was computed in each front at the same time. The crowding distance represents the density of solutions surrounding a particular solution in the objective space. Based on these metrics, a new population is selected for the next generation. After that, a two-point crossover refers to the combination of different individuals into a new one. Mutation forces individuals to change their parameters with a certain probability. This set of processes is repeated for a predefined number of generations. Finally, the population of the final generation describes the Pareto front of the objective space. The parameters for the genetic algorithm are detailed in Table 1. All experiments described in this study were conducted using these parameters.

Algorithm 1: Creating Individual Geofence Candidates

```

[Input] poi: a target point of interest
        r_scan: the scanning range
[Output] i: geofence individual

// determine the center point
radius = uniformRandom(from 0 to r_scan)
angle = uniformRandom(from 0 to 2π)
x = poi.x + radius * cos(angle)
y = poi.y + radius * sin(angle)

// determine the radius
r_max = r_scan - radius
r = uniformRandom(from 0 to r_max)

return i = [x, y, r]
```

3.3.3. User Coverage-Penalized Genetic Algorithm

While the first approach in Section 3.3.2 allows designers to select the preferred solution from the trade-offs, the second approach in this subsection offers an optimal solution by converting the objectives into a single-objective function. To maintain the diversity of the genetic algorithm, user coverage-penalized optimization, the minimization of the location relevance value while treating user coverage as a soft constraint, is considered. This

approach requires a hyperparameter, that is, the minimum user coverage rate cr_{min} . The output geofence is expected to satisfy the number of users covered for the scanning range.

The *UserCoveragePenalizedGA* in Figure 5 illustrates the flow of the proposed optimization method using a genetic algorithm. The basic process is the same as NSGA-II given in Section 3.3.2—create, evaluate, select, take crossover, mutate, and repeat them a predefined number of generations. Finally, the fittest individual of the final generation is the best output and the answer, which is a corresponding *flexible CGF* of the POI within the predefined scanning range r_{scan} . Populations in each generation were created by Algorithm 1. Then, the evaluation for individuals was redesigned for single-objective optimization. The fitness function is depicted by the sum of two terms—a location relevance function, $f(i)$, and a penalty function for the user coverage, $g(i)$.

$$\text{Minimize } F(i) = f(i) + g(i) \quad (5)$$

As stated in Section 3.2.1, the average distance from the boundary of the geofence to the target POI was introduced for the fitness evaluation. Let $dist(poi, i.xy)$ be the Euclidean distance from the location of a POI to the center position of the individual and $i.r$ be the radius of the individual; it can be given as Equation (6).

$$f(i) = \frac{dist(poi, i.xy) + i.r + |dist(poi, i.xy) - i.r|}{2} \quad (6)$$

The penalty function $g(i)$ contributes to user coverage problems by adding a measure of violation of cr_{min} . Let μ be the penalty coefficient and $cr(i)$ be the user coverage rate with respect to a subtrajectory dataset D_{sub} ; then, preserving the function $g(i)$ is given as follows:

$$g(i) = \mu \max(0, cr_{min} - cr(i)) \quad (7)$$

At first, cr_{min} was regarded as a constraint condition to be met, that is, $\mu = \infty$. Excluding geofences that fall below the anticipated number of users from the candidates for solutions leads to the optimization method strictly following developers' intentions. However, a higher cr_{min} reduces the diversity of populations in the genetic algorithm because fewer individuals can satisfy the constraints, resulting in significant time for convergence of the answer. In this study, μ is set as a finite number, and it replaced the optimization problem with an unconstrained problem by the variable penalty. Since the proposed method solves minimization problems, an individual with a lower value is fitter and superior to others. According to the results of evaluation, offspring for the next generation are formed via tournament selection, a two-point crossover, and mutation to keep the diversity of populations and explore a wider range of the search space. The tournament selection involves randomly picking up subsets of individuals, and then the fittest individuals in each subset are preserved. A two-point crossover refers to the combination of different individuals into a new one. Mutation forces individuals to change their parameters with a certain probability. The parameters for the genetic algorithm are detailed in Table 1. All experiments described in this study were conducted using these parameters.

4. Technical Analysis of Geofence Determination Methods

This section evaluates the selection and convergence of solutions to verify the feasibility of each proposed method. Section 4.1 presents the visualization results of the Pareto fronts and discusses the decision-making process for determining optimal geofence parameters. Section 4.2 investigates the statistical deviation in the fittest values under different settings of the minimum user coverage rate and assesses the consistency of the output solutions. The genetic algorithms used in this analysis are executed with parameters detailed in Table 1. The dataset employed is described in Section 3.1.3, and target POIs and the scanning range (equal to 75 m) are consistent with the specifications.

4.1. Visualization of the Pareto Front in Multiobjective Genetic Algorithm

The selection of geofence parameters by the multiobjective genetic algorithm, as introduced in Section 3.3.1, aims to maximize the user coverage and minimize the average POI-notification distance simultaneously. The Pareto fronts, depicted as red lines in Figure 6, demonstrate the trade-off relationships between these objectives for POI (A), (B), and (C). It is considered that the point pattern influences the shape of each Pareto front. Each POI case successfully suggested several viable candidates for geofence parameters.

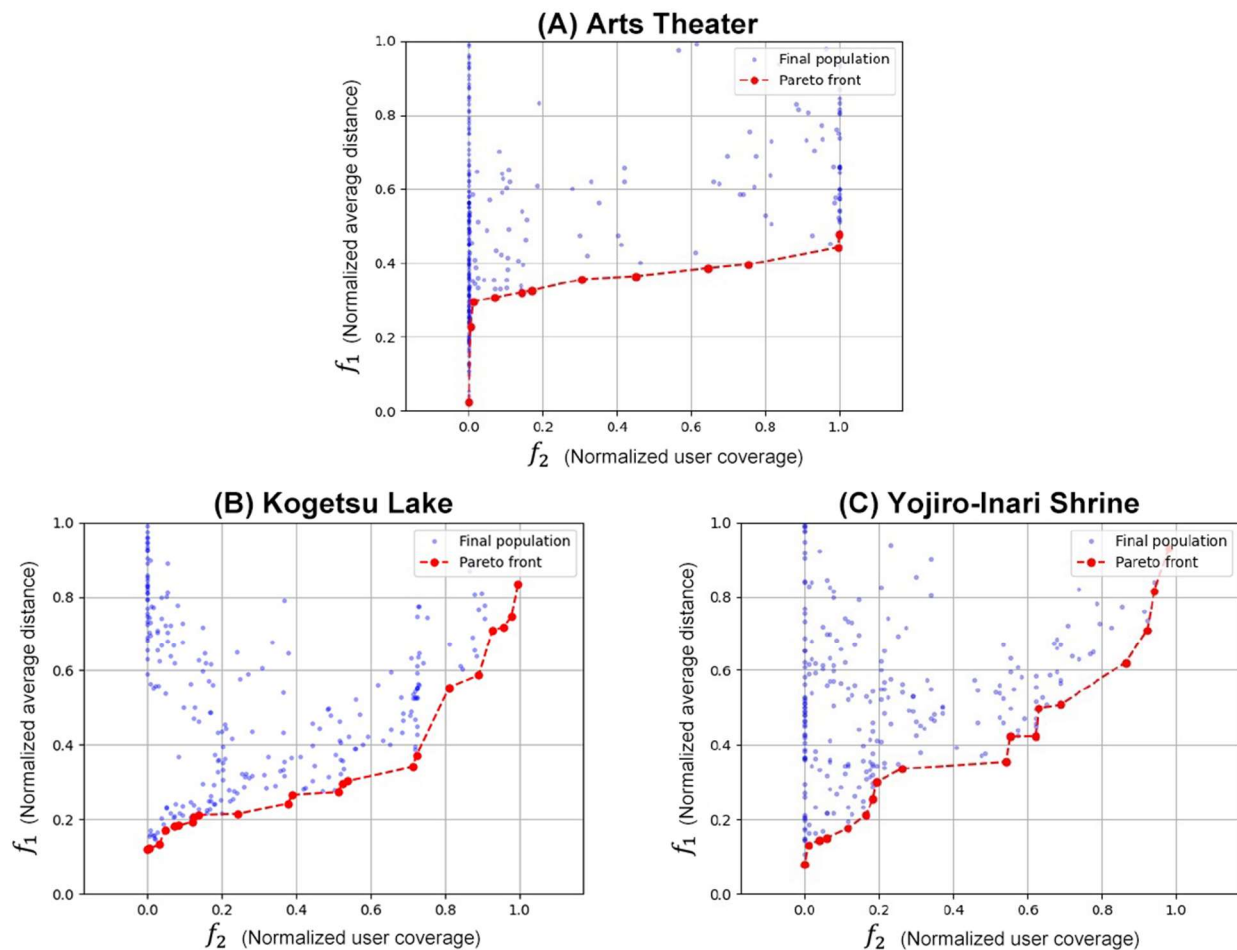


Figure 6. Pareto fronts in three target areas. The trade-offs between user coverage and the approximated average POI-notification distance can be observed in the lines connecting the output red points. Individuals with a normalized average distance exceeding 1.0 in the final population were excluded from the figures because they do not satisfy the solution spaces.

In contrast to the single-objective genetic algorithm discussed in Section 3.3.2, selecting an optimal solution from the Pareto front necessitates a more complex decision-making process. First, integrating specific domain knowledge about the physical environment surrounding the POIs—such as terrain, entry points, and potential obstacles—along with operational considerations like the estimated duration of guide content in audio tour applications can facilitate decision-making. Additionally, deriving a reasonably balanced solution from the Pareto front offers a viable approach to optimizing outcomes. For example, areas with simple clusters, such as POI (A) and (B) in Figure 6, exhibited clearer elbows on their curves. The elbow represents a transition point where the rate of variability changes drastically and is utilized for estimating the number of clusters in point pattern analysis [34]. The elbow points can be identified at $f_2 \approx 1.0$ for POI (A), $f_2 \approx 0.7$ for POI (B), and $f_2 \approx 0.6$ for POI (C). To maintain a reasonable balance between the objectives, these points may be

considered optimal and cost-effective choices. Covering more tourists may lead to a larger degradation of the location relevance, as indicated by the Pareto fronts.

4.2. Stability of Solution Output in User Coverage-Penalized Genetic Algorithm

This subsection examines the stability of the optimization algorithm by analyzing the deviation of the fittest value from the mean after 30 executions of the geofence determination. For POI (A), (B), and (C), the algorithm utilized 150 trajectories that detected entries within each scanning range. The penalty coefficient, $\mu = 2000$, indicates that the penalty function adds an equivalent of 20 m of average distance as a penalty for every 1.0% violation of the user coverage rate limitation. If a high value is set, the user coverage rate of a geofence individual output by this method is strictly limited to the minimum required coverage rate cr_{min} .

Figure 7 presents the results with different minimum user coverages for POI (A), (B), and (C). First, a result of geofence parameters tends to be unstable when stricter limitations of the user coverage rate are set. This feature was seen in the case of POI (B) and (C). Second, when the randomness of the point pattern is lower, the optimal geofence individual has the possibility of output in a more stable way. The areas of POI (B) and (C) have this tendency, especially with $cr_{min} \geq 0.7$. As the point patterns are highly related to the solution space, complex distribution may lead the population to the local optimal solution. Hence, finding appropriate genetic algorithm parameters regarding the probabilities of crossover and mutation becomes challenging for exploration in such areas.

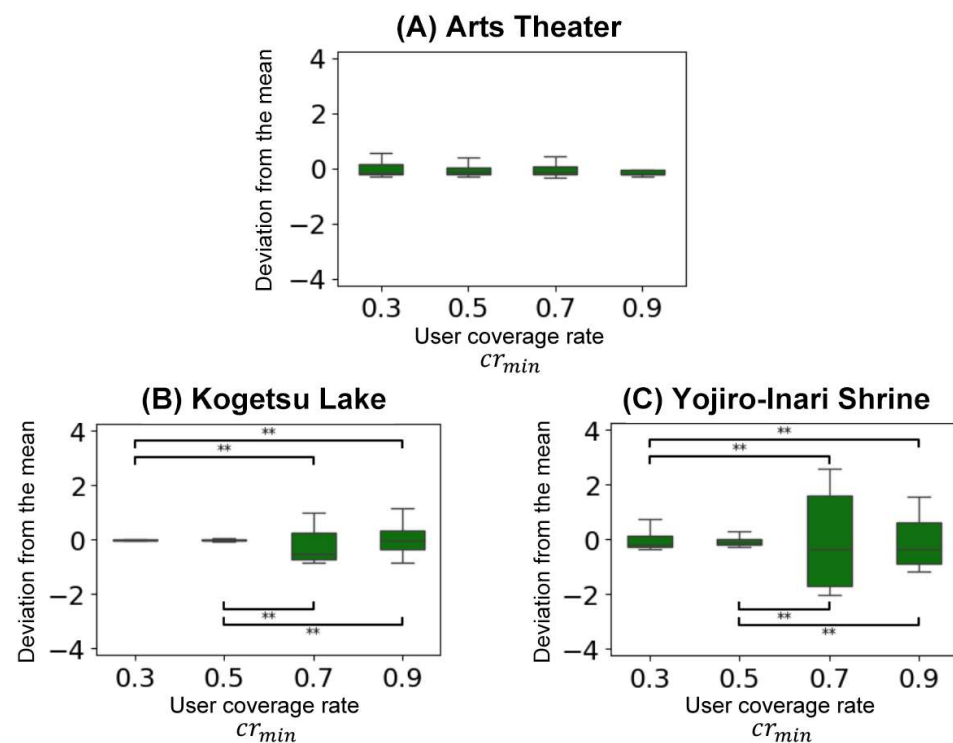


Figure 7. This figure shows the results of the deviation of the fittest value from the mean after 30 trials for each POI at different minimum user coverage rates, suggesting that (1) higher minimum user coverage rates lead to more unstable results and (2) more random point patterns also yield unstable results. In this study, the Fisher's F-test was employed to assess the equality of variances across multiple group pairs. The F-test compares the variances between each pair of groups, and under the null hypothesis, it assumes that the variances within each pair are equal (**: $p < 0.01$).

5. Comparative Analysis of Geofencing Performance Improvement

This section evaluates the impact of the proposed data-driven geofence determination on enhancing the service quality of POI notifiers. Specifically, this study conducts a com-

parative analysis of actual location relevance and user coverage, as measured by test data, between the proposed data-driven method and conventional geofence parameter settings. As described in Section 2.2, geofences established through fully manual processes are straightforward but often encounter issues due to overly large radii. This scenario provides a baseline for comparison to assess improvements offered by alternative parameter settings. Prior research has highlighted that a deep understanding of human mobility patterns and reliability assessments significantly aid service designers in configuring more effective geofences. While these insights are valid, this section argues that the data-driven approach addresses the limitations associated with manual and semi-manual parameter settings and optimizes geofence parameters for superior service quality in terms of the metrics.

5.1. Outline of Experiment

The following five types of geofences were set by different methods: GF-BASE (Geofence as a baseline), GF-NDE (Geofence designed by Non-Domain Expert), GF-DE (Geofence designed by Domain Expert), GF-SIM (Geofence with the reliability score calculated by Simulation), GF-AUTO (Geofence generated by the user coverage-penalized genetic algorithm). GF-BASE refers to the straightforward setting of geofences, namely *fixed* CGFs with a 75 m radius. The three target areas are shown in Section 3.1.3—Arts Theater (the area of POI (A)), Kogetsu Lake (the area of POI (B)), and Yojiro-Inari Shrine (the area of POI (C)). To obtain data for GF-NDE, GF-DE, and GF-SIM, a user survey was conducted with ten subjects divided into two groups. All participants specialized in computer science and understood the concept of geofencing and the problems of GF-BASE. Initially, each person in both groups sets a geofence without any hints, using the geofence editor, which utilizes OpenStreetMap as its base map, creating a GF-NDE. Subsequently, subjects in Group 1 could refer to heatmaps representing pedestrian flows around the areas and update their settings (GF-DE), investigating the influence of knowing tourists' movements on the improvement of geofence settings. Subjects in Group 2 refer to the Geofence Index every time they input geofence parameters. The Geofence Index is the reliability score, primarily calculated by the average proportion between the predefined sampling rate and the road-network-based estimated duration of passing through input geofences [20]. This methodology was implemented, and the effects of the reliability score on geofence designs were compared from the service quality perspectives. In addition to these manual and semi-manual methods, GF-AUTO, generated by the proposed method, was prepared with different numbers of trajectories in the training data. Each type of geofence was evaluated using 200 trajectories generated by the simulator described in Section 3.1.3. The geofences created by five different methods were evaluated using two metrics:

- Actual user coverage—this counts the number of users who have been detected entering the geofence at least once. Even if a user crosses the geofence boundary, there may be instances in which the application does not detect it due to the sampling rate or GPS errors.
- Actual average POI-notification distance—the location relevance in Section 3.2.1 simply calculates the longest and shortest distances between the geofence boundary and the POI to save computational costs. Meanwhile, this scans the actual reference point of each trajectory that enters the geofence for the first time and calculates the distances from these points to the POI.

These are calculated from the test data of the GPS trajectories. To clearly interpret differences in the results along these axes, values were normalized based on the results of GF-BASE.

5.2. Results and Discussion

Figure 8 shows the results of the evaluations. Groups 1 and 2 were separately plotted for three areas. For any target POI, the user coverage rates of GF-AUTO resulted in high values (approximately 1.0). According to the Pareto front outlined in Section 4.1, the average POI-notification distances were likely the most optimal values within the solution

spaces. However, additional data appear to be needed for appropriate geofence settings, especially in areas where pedestrian flows are randomly scattered. For example, in the area of POI (C) for Group 2, GF-AUTO with limited training data could underperform compared to manual settings.

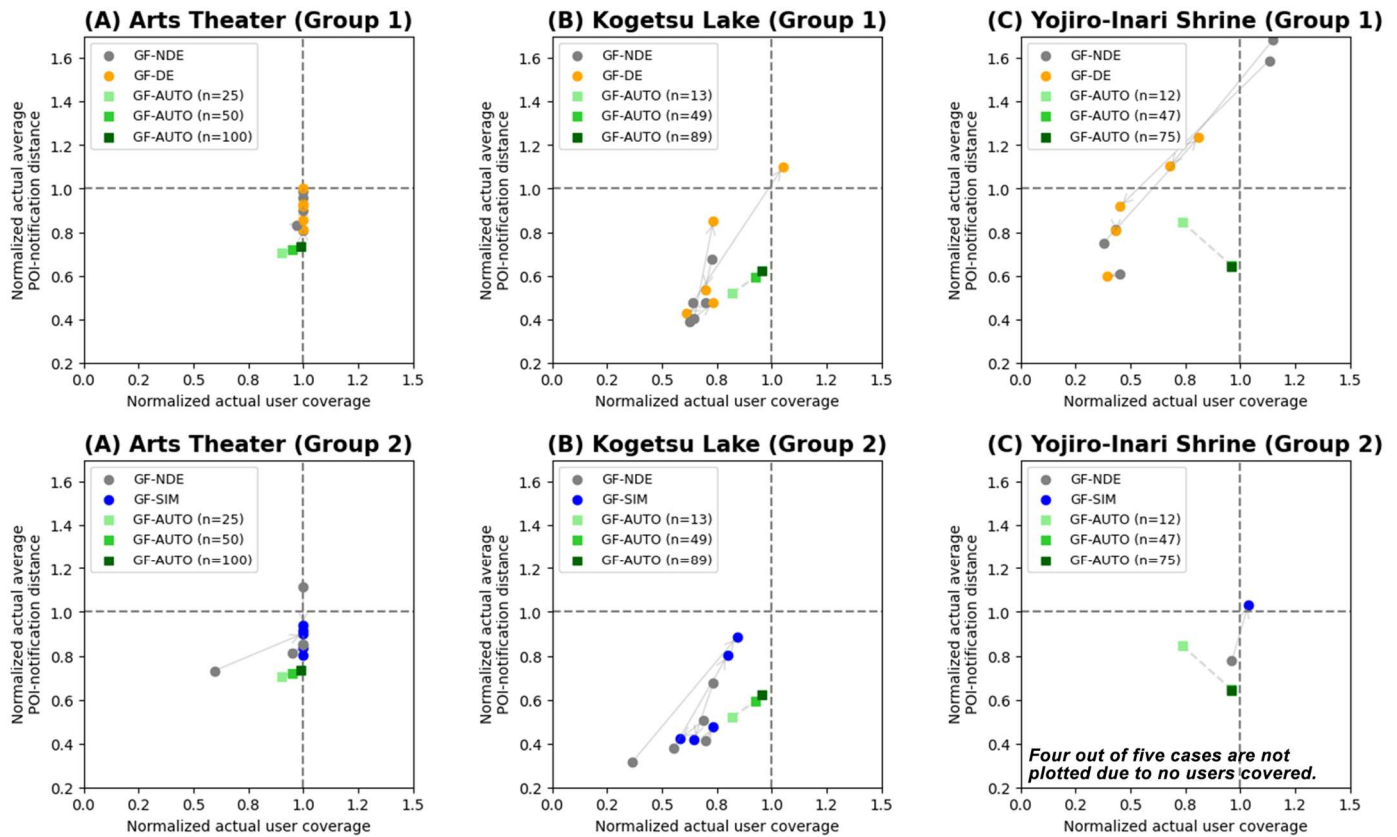


Figure 8. Results of our survey. For each type of geofence, the actual user coverage rate and the average POI-notification distance were evaluated using a test dataset (200 trajectories). Values were normalized based on the results from GF-BASE. The figure consists of six plots arranged in two rows and three columns: the upper row displays the changes in geofence performance set by five subjects in Group 1, and the lower row shows changes set by five subjects in Group 2. Each column corresponds to one of the three POIs.

Few notable differences were found in the results between the groups. The post-user study questionnaires suggested that even base maps enable geofence designers to approximate suitable geofences based on the width and shape of sidewalks. However, a deep understanding of pedestrian flows and reliability simulations may lead designers to set larger geofences than necessary. In some cases of the area of POI (B), the average POI-notification distances for GF-DE or GF-SIM increased compared to GF-NDE. Furthermore, design methods using actual data help prevent the establishment of inappropriate geofences based on preconceptions. In the area of POI (C) for Group 2, four out of five cases covered no pedestrians. Even if footpaths are visible near the POI, walking tourists do not always use them.

Overall, GF-AUTO contributes to geofence optimization based on two objectives. In areas where designers can identify a simple cluster, such as the area of POI (A), GF-AUTO offers an effective tool for optimizing average distances. In areas with complex point patterns, such as the areas of POI (B) and (C), GF-AUTO consistently ensures the high performance of geofences.

6. Conclusions

This study has explored computational methods for generating geofence parameters, enabling scalable and adaptable POI notifiers. The case study on geofence designs revealed a formalization strategy for maximizing location relevance and user coverage. To implement the automatic system of geofence determination, a multiobjective optimization approach and a user coverage-penalized optimization approach were examined using synthetic GPS trajectories that simulate actual movements in Senshu Park, Akita City, Japan. The former approach provides geofence designers with multiple options based on the trade-offs between the objectives, while the latter suggests a single solution based on the minimum user coverage rate requirements. Both approaches offer feasible ways to identify the optimal solutions automatically. Furthermore, the results of this comparative study show that the accumulation of tourists' GPS trajectory data can help to identify better geofence parameters than those achieved through manual and semi-manual settings. This implies that a data-driven approach can overcome technical constraints, such as tracking errors and sampling rates, while considering the dynamic changes in the environment.

Future research is needed to extend the necessity of geofence optimization to diverse geofence services. As this study highlights, different geofencing roles, such as *nearby-geofence* and *staying-geofence* in Section 2.1, may require distinct optimization approaches. In addition, refining the optimization methods for POI notifiers to account for a wider array of geographic and human mobility factors is advisable. The candidates for geofence parameters presented by the Pareto front in Section 4.1 may imply unexplored axes of evaluation that could enhance geofence effectiveness. While many practitioners recognize the applicability of geofencing, the optimal configuration of such systems, particularly in the context of dynamic environmental conditions, remains an underexplored area in location-based services research. The data-driven approaches in this study underscore the potential for addressing these complexities, encouraging further empirical and theoretical work.

To further capitalize on the findings from this study, it is imperative to delve deeper into the interaction between geofence performance and notification delivery efficiency in real-world scenarios. Future research should focus on integrating diverse human behavioral patterns. Establishing a comprehensive theoretical framework that elucidates the relationship between geofence parameters and user interaction could substantially advance the sophistication of geofence services. By pursuing these avenues, research can build upon the current study's groundwork to develop geofence technologies that more precisely reflect and respond to the complexities of human mobility and urban dynamics.

Author Contributions: Conceptualization, Iori Sasaki, Masatoshi Arikawa, Min Lu, Tomihiro Utsumi and Ryo Sato; data curation, Iori Sasaki; formal analysis, Iori Sasaki; funding acquisition, Iori Sasaki, Masatoshi Arikawa and Min Lu; investigation, Iori Sasaki; methodology, Iori Sasaki and Masatoshi Arikawa; project administration, Iori Sasaki, Masatoshi Arikawa and Ryo Sato; resources, Iori Sasaki; software, Iori Sasaki; supervision, Masatoshi Arikawa; validation, Iori Sasaki; visualization, Iori Sasaki; writing—original draft preparation, Iori Sasaki; writing—review and editing, Iori Sasaki, Masatoshi Arikawa, Min Lu, Tomihiro Utsumi and Ryo Sato. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported partly by JSPS KAKENHI Grant Numbers JP22KJ0325, JP22H00764, JP19H04120, JP23K11362, and JP24K15631.

Data Availability Statement: In this study, synthetic data were created and used, and the method for creating this data is detailed in Appendix A. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors would like to thank the participants in the user experiment.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

The experimental analyses in this study have introduced massive amounts of synthetic GPS trajectories for comprehensive technical surveys. Appendix A details a method for generating synthetic GPS trajectory datasets from road network data of OpenStreetMap. This comprises three stages: (1) footpath extraction, (2) random route selection, and (3) trajectory generation. The generated trajectory dataset is demonstrated in Figure 3.

The first step involves extracting a network of walking paths in Senshu Park, Akita City, Japan, from OpenStreetMap. OpenStreetMap provides a map resource for various pathway types, including those designated for pedestrian use. In the extracted network, nodes represent intersections or endpoints of paths, while edges are defined as the connections between these nodes. Each edge carries a geometry attribute that defines the shape of the footpath using a polyline feature—a sequence of coordinates. The length of each polyline is utilized as the edge cost in this network, influencing the route selection algorithms described in the random route selection. These geometry data are used to trace segments of the simulated route when generating trajectories.

The simulation generates trajectories that pass through various routes. Based on the obtained network, cost scaling and Dijkstra's algorithm were applied to determine the travel route of each trajectory. Initially, the cost of each edge was scaled by randomly updating its value to anywhere from half to double before selecting a walking route. Subsequently, Dijkstra's algorithm was used on this cost-updated network to select a walking route. In geographic information systems, Dijkstra's algorithm, one of the most popular algorithms, serves as a fundamental function to find the shortest path between nodes, considering cost distances. The algorithm begins at the starting node and explores all neighboring nodes at the current depth level before proceeding to nodes at the next depth level. For all trajectories in this dataset, the starting and the goal nodes were fixed, and the process of cost scaling and selecting the shortest path was repeated.

Finally, the simulation generates trajectories that imitate the actual behavior of GPS tracking. Three kinds of hyperparameters were set: the recording interval, walking speed, and GPS noise. The recording interval was set to 8.0 s. Considering various walker characteristics, such as gender and age, the walking speed of each trajectory varied following a normal distribution, with an average speed of 1.3 m per second and a standard deviation of 0.2 m per second. The recording interval and walking speed determine the distance intervals of reference points in a GPS trajectory. Essentially, the simulator generates a synthetic GPS trajectory along the skeleton shape of the selected route and the distance interval. In addition, GPS noise was considered. In this dataset, the position of each reference point of a trajectory was randomly reset according to a normal distribution, with the original position as the average and 5.0 m as the standard deviation.

References

- Garzon, S.R.; Deva, B. Geofencing 2.0: Taking location-based notifications to the next level. In Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Seattle, WA, USA, 13–17 September 2014; pp. 921–932. [CrossRef]
- Huang, H.; Gartner, G.; Krisp, J.M.; Raubal, M.; Weghe, N.V. Location based services: Ongoing evolution and research agenda. *J. Locat. Based Serv.* **2018**, *12*, 63–93. [CrossRef]
- Ullah, F.; Haq, H.U.; Khan, J.; Safeer, A.A.; Asif, U.; Lee, S. Wearable IoTs and geo-fencing based framework for COVID-19 remote patient health monitoring and quarantine management to control the pandemic. *Electronics* **2021**, *10*, 2035. [CrossRef]
- Cheng, G.; Guo, Y.; Chen, Y.; Qin, Y. Designating city-wide collaborative geofence sites for renting and returning dock-less shared bikes. *IEEE Access* **2019**, *7*, 35596–35605. [CrossRef]
- Apple Developer Documentation | Managing Location-Based Reminders. Available online: https://developer.apple.com/documentation/eventkit/managing_location-based_reminders (accessed on 21 February 2024).
- Stevens, M.; Atkins, E. Geofence definition and deconfliction for UAS traffic management. *IEEE Trans. Intell. Transp. Syst.* **2021**, *22*, 5880–5889. [CrossRef]
- Sheppard, J.K.; McGann, A.; Lanzone, M.; Swasgood, R.R. An autonomous GPS geofence alert system to curtail avian fatalities at wind farms. *Anim. Biotelemetry* **2015**, *3*, 43. [CrossRef]
- Garcia, M.B. Location-based marketing using mobile geofencing: Lessons learned from a user-centered application development research. *Int. J. Technol. Mark. (IJTMKT)* **2022**, *17*, 1–29. [CrossRef]

9. Xie, Y.; Zhang, Z.; Liu, Y.; Chen, E.; Li, N. Evaluation method of IP geolocation database based on city delay characteristics. *Electronics* **2024**, *13*, 15. [\[CrossRef\]](#)
10. Garg, A.; Choudhary, S.; Bajaj, P.; Agrawal, S.; Kedia, A.; Agrawal, S. Smart geo-fencing with location sensitive product affinity. In Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Redondo Beach, CA, USA, 7–10 November 2017; pp. 1–10. [\[CrossRef\]](#)
11. Bareth, U.; Kupper, A.; Freese, B. Geofencing and background tracking—The next features in LBS. In Proceedings of the 41st Annual Conference of the Gesellschaft für Informatik e.V. (INFORMATIK 2011), Berlin, Germany, 4–7 October 2011.
12. Android Developer. Create and Monitor Geofences. Available online: <https://developer.android.com/training/location/geofencing/> (accessed on 21 February 2024).
13. Apple Developer. Monitoring the User's Proximity to Geographic Regions. Available online: https://developer.apple.com/documentation/corelocation/monitoring_the_user_s_proximity_to_geographic_regions/ (accessed on 21 February 2024).
14. Loyola, L.; Wong, F.; Pereira, D.; Sanson, H. Extending battery lifetime of mobile devices with geofence capabilities on dynamic-speed urban environments. In Proceedings of the Second ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems, Dallas/Fort Worth, TX, USA, 4 November 2013; pp. 51–58. [\[CrossRef\]](#)
15. Tang, S.; Yu, Y.; Zimmermann, R.; Obana, S. Efficient geo-fencing via hybrid hashing: A combination of bucket selection and in-bucket binary search. *ACM Trans. Spat. Algorithms Syst.* **2010**, *1*, 5. [\[CrossRef\]](#)
16. Li, S.; Sun, W.; Song, R.; Shan, Z.; Zhang, X. Quick geo-fencing using trajectory partitioning and boundary simplification. In Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Orlando, FL, USA, 5–8 November 2013; pp. 580–583. [\[CrossRef\]](#)
17. Stevens, M.N.; Rastgoftar, H.; Atkins, E.M. Geofence Boundary Violation Detection in 3D Using Triangle Weight Characterization with Adjacency. *J. Intell. Robot. Syst.* **2019**, *95*, 239–250. [\[CrossRef\]](#)
18. Bareth, U.; Kupper, A.; Ruppel, P. geoXmart—A marketplace for geofence-based mobile services. In Proceedings of the 2010 IEEE 34th Annual Computer Software and Applications Conference, Seoul, Republic of Korea, 19–23 July 2010; pp. 101–106. [\[CrossRef\]](#)
19. Velkova, V.; Ivanov, R. Mobile application for creating and exporting geofences. In Proceedings of the 2023 International Conference Automatics and Informatics (ICAI), Varna, Bulgaria, 5–7 October 2023; pp. 221–224. [\[CrossRef\]](#)
20. Garzon, S.R.; Arbuzin, D.; Kupper, A. Geofence Index: A performance estimator for the reliability of proactive location-based services. In Proceedings of the 2017 18th IEEE International Conference on Mobile Data Management (MDM), Daejeon, Republic of Korea, 29 May–1 June 2017; pp. 1–10. [\[CrossRef\]](#)
21. Sasaki, I.; Arikawa, M.; Lu, M.; Sato, R.; Utsumi, T. Adaptable data-driven geofences for notifying points of interest using tourists' GPS trajectories. In Proceedings of the 7th ACM SIGSPATIAL Workshop on Location-based Recommendations, Geosocial Networks and Geoadvertising, Hamburg, Germany, 13 November 2023; pp. 37–43. [\[CrossRef\]](#)
22. Garzon, S.R.; Deva, B. On the evaluation of proactive location-based services. In Proceedings of the 2015 IEEE 39th Annual Computer Software and Applications Conference, Taichung, Taiwan, 1–5 July 2015; pp. 585–594. [\[CrossRef\]](#)
23. Statler, S. Geofencing: Everything you need to know. In *Beacon Technologies*; Apress: Berkeley, CA, USA, 2016. [\[CrossRef\]](#)
24. Su, X.; Khoshgoftaar, T.M. A survey of collaborative filtering techniques. *Adv. Artif. Intell.* **2009**, *2009*, 421425. [\[CrossRef\]](#)
25. Basid, P.M.N.S.A.; Tolle, H.; Ramdani, F. Designing module e-complaint system based on geotagging and geo-fencing. *Int. J. Interact. Mob. Technol.* **2017**, *11*, 113–129. [\[CrossRef\]](#)
26. Tokita, H.; Arikawa, M.; Si, R.; Sato, R.; Tayama, R.; Takahashi, A. Virtual circular geofences for points and regions of interests with spatial context. *Abstr. ICA* **2019**, *1*, 366. [\[CrossRef\]](#)
27. Parent, C.; Spaccapietra, S.; Renso, C.; Andrienko, G.; Andrienko, N.; Bogorny, V.; Damiani, M.L.; Divanis, A.G.; Macedo, J.; Pelekis, N.; et al. Semantic trajectories modeling and analysis. *ACM Comput. Surv.* **2013**, *45*, 42. [\[CrossRef\]](#)
28. Zheng, Y.; Xie, X.; Ma, W.Y. GeoLife: A collaborative social networking services among user, location and trajectory. *IEEE Data Eng. Bull.* **2010**, *33*, 32–39.
29. Wang, Y.; Zhang, D.; Liu, Q.; Shen, F.; Lee, L.H. Towards enhancing the last-mile delivery: An effective crowd-tasking model with scalable solutions. *Transp. Res. E Logist. Transp. Rev.* **2016**, *93*, 279–293. [\[CrossRef\]](#)
30. Vu, T.; Migliorini, S.; Eldawy, A.; Belussi, A. Spatial Data Generators. In *Spatial Gems*; Krumm, J., Züfle, A., Shahabi, C., Eds.; Association for Computing Machinery: New York, NY, USA, 2022; Volume 1, pp. 13–23. [\[CrossRef\]](#)
31. OpenStreetMap. Available online: <https://www.openstreetmap.org/> (accessed on 21 February 2024).
32. Rainville, F.M.D.; Fortin, F.A.; Gardner, M.A.; Parizeau, M.; Gagné, C. DEAP: A python framework for evolutionary algorithms. In Proceedings of the 14th Annual Conference Companion on Genetic and Evolutionary Computation, Philadelphia, PA, USA, 7–11 July 2012; pp. 85–92. [\[CrossRef\]](#)
33. Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput.* **2014**, *18*, 577–601. [\[CrossRef\]](#)
34. Wang, S.; Sun, W. Capacity value assessment for a combined power plant system of new energy and energy storage based on robust scheduling rules. *Sustainability* **2023**, *15*, 15327. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.